

Introduction To The ASP+ Page Framework, Part 1 (Web Forms)

**Susan Warren
Program Manager
.NET Frameworks
Microsoft Corporation**

4-213

Microsoft®
PDC 2000
Professional Developers Conference

Microsoft®
.net

the defining

point

ASP+ Page Framework

- **ASP+ Pages have “.ASPX” extension**
 - Side-by-side support with classic ASP
- **Compiled to native code**
 - All languages, including “script”
 - Still “Just hit save” development
- **New control-based, event-driven execution model**
 - Much less code required
 - Cleanly encapsulated functionality

Agenda, Parts 1 & 2

■ Part 1

- Page Basics
- Server Control Basics
- Code Basics
- Control Families
- Page controls (“pagelets”)

■ Part 2

- Input Validation
- Targeting Uplevel Browsers
- Data Binding
- Data Binding With Templates

Page Basics

ASP+ Page Syntax

- **Most basic page is just static text**
 - Any HTML page can be renamed .ASPX
- **Pages may contain:**
 - **Directives:** `<%@ Directive %>`
 - **Server controls:** `<tag runat=server>`
 - **Code blocks:** `<script runat=server>`
 - **Data bind expressions:** `<%= # %>`
 - **Server side comments:** `<%- - - - %>`
 - **Render code:** `<%= %>` and `<% %>`

Server Control Basics



Server Control Syntax

- Declared as HTML or XML tag with **runat=server** attribute

- Examples:

```
<input type=text id=text2 runat=server>  
<asp:calendar id=myCal runat=server/>
```

- Tag identifies control to create
 - Control is implemented as NGWS class
- ID attribute provides programmatic accessor

Server Control Properties

- Tag attributes map to control properties

```
<tag id="t1" Text="Foo" runat=server>
```

```
<tag id="t2" Height="5" runat=server>
```

- Control properties can be set programmatically

```
t1.Text = "Bar"
```

```
t2.Height = 99
```

- Tags, attributes case insensitive

Input Control Postback

- **Input controls can automatically populate their object model from values entered on the client**
- **No “glue” code required to restore values to inputs**
- **No state stored on the server!**
- **Works with all browsers!**

Code Basics

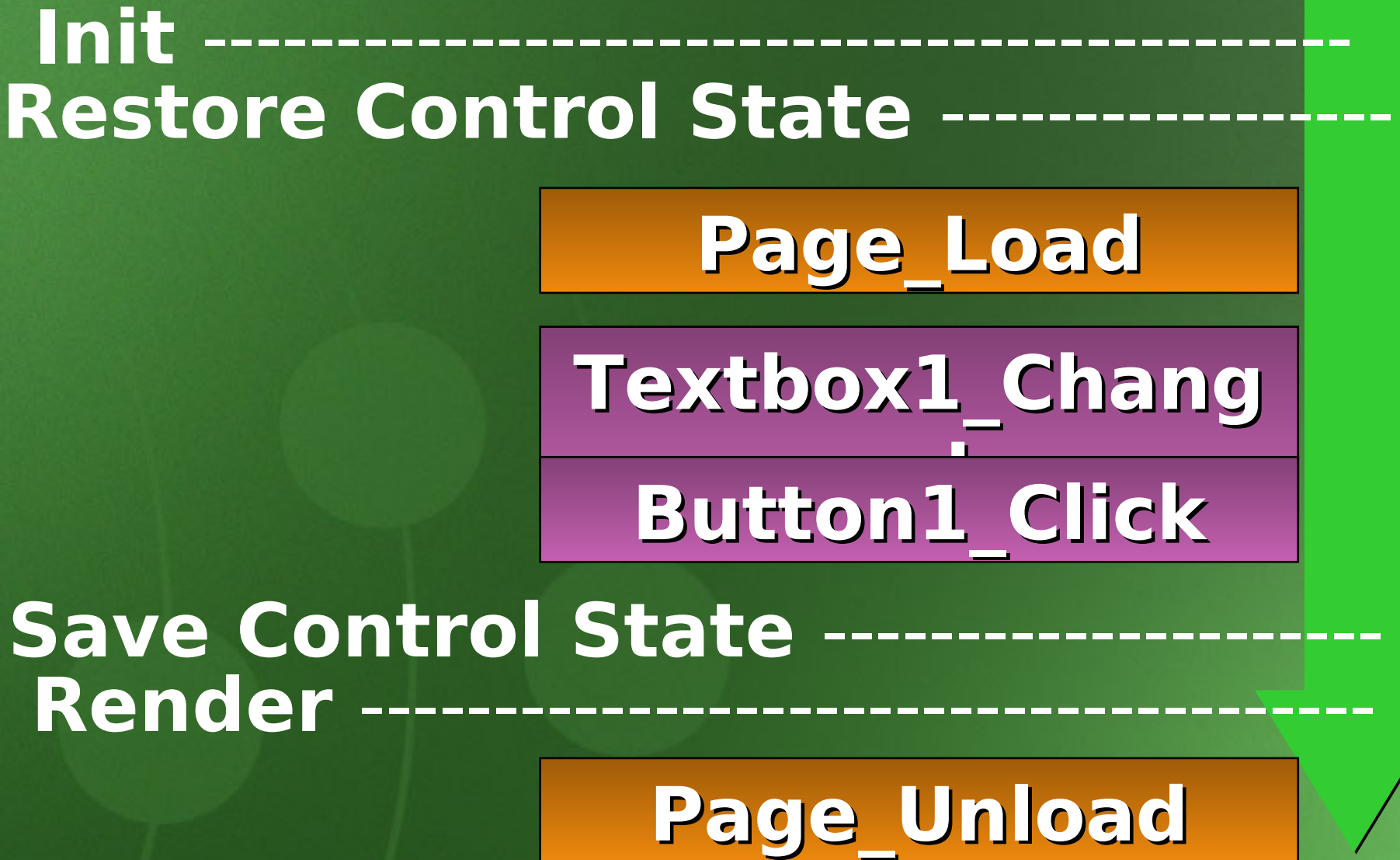
Server Code Block

- **Server code lives in a script block marked runat=server**
`<script language="VB" runat=server>`
`<script language="C#" runat=server>`
`<script language="JScript" runat=server>`
- **Can contain:**
 - **Variables, Methods, Event Handlers**

Page Events

- Pages can be structured using events
 - Enables clean code organization
 - Avoids the “Monster If” statement
- Code can sync Page events
 - Page_Load
 - Page_Unload
- Code can sync control events
 - Button1_Click
 - Textbox1_Changed

Page Life Cycle



Hooking the Life Cycle Events

Control Events

1. Change Events

2. Action Events

Page_Load

Textbox1_Changed

Button1_Click

Page_Unload



Page_Load Details (1 Of 2)

- Fires at beginning of request after controls are initialized
 - Input controls values already populated

- **Example:**

```
Sub Page_Load
```

```
    message.Text = textbox1.Text
```

```
End Sub
```

Page_Load Details (2 Of 2)

- Page_Load fires on every request
 - Use Page.IsPostBack to execute conditional logic

- Example:

```
Sub Page_Load(s As Object,e As EventArgs)
```

```
    If Not Page.IsPostBack Then
```

```
        'executes only on initial page load
```

```
        Message.Text = "initial value"
```

```
    End If
```

```
    ' Rest of procedure executes on every request
```

```
End Sub
```

Server Control Events

- **Change events**
 - Execute only on next request to server
 - Textbox1_Changed, CheckBox1_CheckChanged, etc.
- **Action events**
 - Causes an immediate postback to server
 - Button1_Click
- **Works with any browser**
 - No client script, no applets,

Control Event Wire-Up

- **Control event handlers are identified on the tag:**

```
<asp:button onclick="button1_click"
runat=server>
```

```
<asp:textbox onchange="text1_changed"
runat=server>
```

- **Event handler code:**

```
Sub btn1_Click(s as Object,e as EventArgs)
Message.Text = "Button1 clicked"
End Sub
```

Page_Unload Details

- Fires after the page is rendered
- Useful for logging and clean up
- Example:

```
Sub Page_Unload(s As Object,e As EventArgs)  
    MyApp.LogPageComplete()  
End Sub
```


ViewState

- **Non-input controls can maintain their data across multiple postback requests**
 - Implemented using a client hidden field
 - Examples: Label, DataGrid, etc.
- **Can be disabled per control or entire Page**
 - Set **maintainstate=false**
 - *4-326 ASP+ Performance and Caching*

<%@ Import %>

Directive

- **Adds code namespace reference**

to page

- **Avoids having to full-qualify .NET types and class names**

```
<%@ Import Namespace="System.Data" %>
```

```
<%@ Import Namespace="System.Net" %>
```

```
<%@ Import Namespace="System.IO" %>
```

Control Families



Server Control Families

- **ASP+ ships 45 built-in controls**
 - **Organized into logical families**
- **HTML Controls**
 - **Controls / properties map 1:1 with HTML**
- **Web Controls**
 - **Intrinsic Controls**
 - **List Controls**
 - **Rich Controls**
 - **Validation Controls**

HTML Server Controls

- Works well with HTML designers
- Properties map 1:1 with HTML
`table.bgcolor = "red"`
- Supported Controls:
 - `<a>`, ``, `<form>`
 - `<table>`, `<tr>`, `<td>`, `<th>`
 - `<select>`, `<textarea>`, `<button>`
 - `<input type=text>`, `<input type=file>`,
 - `<input type=submit>`, `<input type=button>`,
 - `<input type=reset>`, `<input`

Web Controls

- **Consistent object model**

 - `Label1.BackColor = Color.Red`

 - `Table.BackColor = Color.Blue`

- **Abstract / Richer Functionality**

 - **Example: Label**

- **Automatic Uplevel/Downlevel Support**

 - **Example: Validation Controls**

- **Strongly-typed**

 - **Enables better compiler type checking**

Intrinsic Controls

- **Supported Controls:**
 - **<asp:button>, <asp:imagebutton>**
 - **<asp:linkbutton>, <asp:hyperlink>**
 - **<asp:textbox>, <asp:checkbox>**
 - **<asp:radiobutton>, <asp:image>**
 - **<asp:label>, <asp:panel>**
 - **<asp:table>**

List Controls

- **Controls that handle repetition**
 - **Expose templates for item customization**
 - **More about templates in Part 2...**
- **Supported Controls:**
 - **<asp:dropdownlist>, <asp:listbox>**
 - **<asp:radiobuttonlist>, <asp:checkboxlist>**
 - **<asp:repeater>**
 - **<asp:datalist>**
 - **<asp:datagrid>**

Rich Controls

- High-level custom controls
- Supported Controls:
 - `<asp:calendar>`
 - `<asp:adrotator>`
- More will be added in subsequent betas...

Validation Controls

- **Enable client and server validation**
 - Automatically detects uplevel/downlevel
- **Supported Controls:**
 - `<asp:RequiredFieldValidator>`
 - `<asp:RangeValidator>`
 - `<asp:CompareValidator>`
 - `<asp:RegularExpressionValidator>`
 - `<asp:CustomValidator>`

More Controls Samples

- **ASP+ QuickStart in .NET Framework SDK**
 - **Web Forms Controls Reference**

<http://localhost/quickstart/aspplus/>

Other Controls

- **Future beta: Mobile Controls**
 - Can emit WML or small device HTML
 - Cell Phones, Pagers, PDAs
 - *4-324 Targeting Mobile Devices with ASP+*
- **Custom compiled controls**
 - Easy for 3rd parties to create controls
 - *4-428 Building ASP+ Server Controls #1*
 - *4-429 Building ASP+ Server Controls #2*
 - *1-431 Leveraging IE 5.5 w/ ASP+ Server Controls*
- **Pagelets**
 - Design using same techniques as pages

Page Controls ("Pagelets")

Pagelets

- **Simple way for page developers to author controls**
- **Enables full encapsulation**
 - Supports nested controls
 - Separate code namespace
 - Separate code language
- **Great way to partition work across multiple developers**
- **Great way to reuse work across multiple pages and applications**

<%@ Register %>

Directive

- Registers pagelet for use on a page:

```
<%@ Register TagPrefix="Acme"  
    TagName="Message" Src="pagelet1.aspx" %>
```

```
<Acme:Message runat="server"/>
```

Exposing An Object Model

- Pagelets can expose object model to pages
 - Properties, Fields, Events, Methods

- Example:

```
<script language="VB" runat="server">  
    Public Color As String = "blue"  
</script>
```

```
<font color=<%=Color%>>  
    This is a simple message pagelet  
</font>
```

Using Pagelets Programmatically

- **Page.LoadControl(string source)**
 - Enables dynamically instantiating new instances of a pagelet
- **Get and customize an instance:**

```
foo = Page.LoadControl("foo.aspx")  
foo.color = "red"
```
- **Insert into the control hierarchy:**

```
myPanel.Controls.Add (foo)
```

Summary

- **ASP+ provides an easy, powerful way to build Web UI that targets any browser**
 - **Simpler development**
 - **Cleaner code**
 - **Better code reuse**
 - **Better language support**
- **Tons of new features here....**
 - **Important to note that developers are not required to use/master everything**

Related Sessions

- **Other PDC sessions recommended:**
 - ***4-428 Building ASP+ Server Controls #1***
 - ***4-429 Building ASP+ Server Controls #2***
 - ***1-431 Leveraging IE 5.5 From ASP+ Server Controls***
 - ***4-324 Targeting Mobile Devices w/ ASP+***
 - ***4-322 ASP+ Tips and Tricks***
 - ***4-334 ASP to ASP+ Migration***

References

- **“A Preview of ASP+” Wrox Press**
 - Chapters 1, 2, 4
- **ASP+ Quickstart in .NET Framework SDK**

Questions?

The background is a solid green color with a subtle gradient. In the lower-left quadrant, there are three faint, light-green circles of varying sizes, connected by thin, curved lines, creating a decorative, abstract pattern.

Where do **you** want to go today?

Microsoft

Backup

